

Extending the Sparkle Core language with object

Zoltán Horváth, Tamás Kozsik and Máté Tejfel

Sparkle is a theorem prover specially constructed for the functional programming language Clean. In a pure functional language like Clean the values of the functional variables are constants; variables of functional programs do not change in time. Hence it seems that temporality has no meaning in functional programs. However, in certain cases (e.g. in interactive or distributed programs, or in ones that use IO), we can consider a series of values computed from each other as different states of the same “abstract object”. For this abstract object we can prove temporal properties. In this paper we present a method to describe abstract objects and temporal properties in an extended version of the Sparkle Core language. The creation of such descriptions will be supported by a refactoring tool. The descriptions are completely machine processable, and provide a way to automatize the proof of temporal properties of Clean programs with the extended Sparkle system.

References

- [1] Achten, P., Plasmeijer, R.: Interactive Objects in Clean. *Proceedings of Implementation of Functional Languages, 9th International Workshop, IFL'97* (K. Hammond et al (eds)), St. Andrews, Scotland, UK, September 1997, LNCS 1467, pp. 304–321.
- [2] Chandy, K. M., Misra, J.: *Parallel program design: a foundation*. Addison-Wesley, 1989.
- [3] Horváth Z., Achten, P., Kozsik T., Plasmeijer, R.: Verification of the Temporal Properties of Dynamic Clean Processes. *Proceedings of Implementation of Functional Languages, IFL'99*, Lochem, The Netherlands, Sept. 7–10, 1999. pp. 203–218.
- [4] de Mol, M., van Eekelen, M., Plasmeijer, R.: Theorem Proving for Functional Programmers, Sparkle: A Functional Theorem Prover, Springer Verlag, LNCS 2312, p. 55 ff., 2001.
- [5] Horváth Z. - Kozsik T. - Tejfel M.: Proving Invariants of Functional Programs. *Proceedings of Eighth Symposium on Programming Languages and Software Tools*, Kuopio, Finland, June 17-18, 2003., pp. 115-126
- [6] Diviánszky P. - Szabó-Nacsa R. - Horváth Z.: A Framework for Refactoring Clean Programs. *6th International Conference on Applied Informatics*, Eger, Hungary January 27-31 2004.
- [7] Horváth Z. - Kozsik T. - Tejfel M.: Verifying invariants of abstract functional objects—a case study. *6th International Conference on Applied Informatics*, Eger, Hungary January 27-31 2004.